FINAL SEMESTER EXAMINATION /
Academic Year 2020 – 2021 / 2nd Semester
Subject : Database Systems
Lecturer : Zain Saifullah
Study Program: Information Technology / IT4
Date of Exam : May 3, 2021

MULYA FAJAR NINGSIH ALWI
001202000101
IT 2020 CLASS 4

Instructions to Students

1. Due date of this examination is **Monday May 10, 2021 23.59 PM**
2. Sanctions will be given to those students who are not following the examination rules
3. All answers to be written directly following the questions. The number to the corresponding question must be written correctly
4. This is a take home examination
5. Students are not allowed to communicate or to cooperate each other or copy someone's work while the examination is going on
6. You must submit your handwritten answer by screenshot it and inserted into word or pdf files. You also must submit screenshoot all answers (from cmd) in the same file. Your file name is YourName_YourStudentID.doc(x) or pdf

---

Questions:

1. (50 Marks) Create SQL Command which

a) Creating the database and tables (at least 4 and has connections among the tables) with the following topics (must be not the same with question no.1 for all students)

Last Digit of your ID	Topics
0	Sports
1	Politics
2	Finance
3	Culinary
4	Plant Production
5	Health
6	Education
7	Transportation
8	Culture
9	Property

Question b) until g) will be done using no a) and also make expalanations on each question

```
XAMPP for Windows - mysql  -u root

Setting environment for using XAMPP for Windows.
Lenovo Flex@NINISMR c:\xampp
# mysql -u root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 13
Server version: 10.4.17-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> CREATE database politics;
Query OK, 1 row affected (0.023 sec)

MariaDB [(none)]> USE politics;
Database changed
MariaDB [politics]> CREATE table province (
    -> code int NOT NULL,
    -> province_name varchar(20),
    -> PRIMARY KEY (code));
Query OK, 0 rows affected (0.044 sec)

MariaDB [politics]> DESC province;
+---------------+------------+------+-----+---------+-------+
| Field         | Type       | Null | Key | Default | Extra |
+---------------+------------+------+-----+---------+-------+
| code          | int(11)    | NO   | PRI | NULL    |       |
| province_name | varchar(20)| YES  |     | NULL    |       |
+---------------+------------+------+-----+---------+-------+
2 rows in set (0.007 sec)

MariaDB [politics]> CREATE table district (
    -> province_code int NOT NULL,
    -> code int(2) zerofill NOT NULL,
    -> district_name varchar(20),
    -> PRIMARY KEY (code),
    -> FOREIGN KEY (province_code)
    -> REFERENCES province(code));
Query OK, 0 rows affected (0.071 sec)

MariaDB [politics]> DESC district;
+---------------+--------------------------+------+-----+---------+-------+
| Field         | Type                     | Null | Key | Default | Extra |
+---------------+--------------------------+------+-----+---------+-------+
| province_code | int(11)                  | NO   | MUL | NULL    |       |
| code          | int(2) unsigned zerofill | NO   | PRI | NULL    |       |
| district_name | varchar(20)              | YES  |     | NULL    |       |
+---------------+--------------------------+------+-----+---------+-------+
3 rows in set (0.035 sec)

MariaDB [politics]> CREATE table sub_district (
    -> district_code int(2) zerofill NOT NULL,
    -> code int(2) zerofill NOT NULL,
    -> sub_district_name varchar(20),
    -> PRIMARY KEY (code),
    -> FOREIGN KEY (district_code)
    -> REFERENCES district(code));
Query OK, 0 rows affected (0.051 sec)

MariaDB [politics]> DESC sub_district;
+------------------+--------------------------+------+-----+---------+-------+
| Field            | Type                     | Null | Key | Default | Extra |
+------------------+--------------------------+------+-----+---------+-------+
| district_code    | int(2) unsigned zerofill | NO   | MUL | NULL    |       |
| code             | int(2) unsigned zerofill | NO   | PRI | NULL    |       |
| sub_district_name| varchar(20)              | YES  |     | NULL    |       |
+------------------+--------------------------+------+-----+---------+-------+
3 rows in set (0.043 sec)

MariaDB [politics]> CREATE table village (
    -> sub_district_code int(2) zerofill NOT NULL,
    -> code int NOT NULL,
    -> village_name varchar(20),
    -> PRIMARY KEY (code),
    -> FOREIGN KEY (sub_district_code)
    -> REFERENCES sub_district(code));
Query OK, 0 rows affected (0.017 sec)

MariaDB [politics]> DESC village;
+------------------+--------------------------+------+-----+---------+-------+
| Field            | Type                     | Null | Key | Default | Extra |
+------------------+--------------------------+------+-----+---------+-------+
| sub_district_code| int(2) unsigned zerofill | NO   | MUL | NULL    |       |
| code             | int(11)                  | NO   | PRI | NULL    |       |
| village_name     | varchar(20)              | YES  |     | NULL    |       |
+------------------+--------------------------+------+-----+---------+-------+
3 rows in set (0.009 sec)

MariaDB [politics]> CREATE table party (
    -> code int(2) zerofill NOT NULL,
    -> party_name varchar(15),
    -> established_year int(4),
    -> party_chairman char(30),
    -> participation_in_legislative_general_election int,
    -> PRIMARY KEY (code));
Query OK, 0 rows affected (0.061 sec)

MariaDB [politics]> DESC party;
+-----------------------------------------------+--------------------------+------+-----+---------+-------+
| Field                                         | Type                     | Null | Key | Default | Extra |
+-----------------------------------------------+--------------------------+------+-----+---------+-------+
| code                                          | int(2) unsigned zerofill | NO   | PRI | NULL    |       |
| party_name                                    | varchar(15)              | YES  |     | NULL    |       |
| established_year                              | int(4)                   | YES  |     | NULL    |       |
| party_chairman                                | char(30)                 | YES  |     | NULL    |       |
| participation_in_legislative_general_election | int(11)                  | YES  |     | NULL    |       |
+-----------------------------------------------+--------------------------+------+-----+---------+-------+
5 rows in set (0.055 sec)
```

```
MariaDB [politics]> CREATE table legislative_candidate (
    -> ID_card_number char(16) NOT NULL,
    -> candidate_name varchar(30),
    -> gender varchar(10),
    -> address varchar(50),
    -> village_code int NOT NULL,
    -> party_code int(2) zerofill NOT NULL,
    -> party_name varchar(15),
    -> serial_number int(2) zerofill NOT NULL,
    -> PRIMARY KEY (serial_number),
    -> FOREIGN KEY (village_code)
    -> REFERENCES village(code),
    -> FOREIGN KEY (party_code)
    -> REFERENCES party(code));
Query OK, 0 rows affected (0.053 sec)

MariaDB [politics]> DESC legislative_candidate;
+---------------+-------------------------+------+-----+---------+-------+
| Field         | Type                    | Null | Key | Default | Extra |
+---------------+-------------------------+------+-----+---------+-------+
| ID_card_number | char(16)                | NO   |     | NULL    |       |
| candidate_name | varchar(30)             | YES  |     | NULL    |       |
| gender        | varchar(10)             | YES  |     | NULL    |       |
| address       | varchar(50)             | YES  |     | NULL    |       |
| village_code  | int(11)                 | NO   | MUL | NULL    |       |
| party_code    | int(2) unsigned zerofill | NO   | MUL | NULL    |       |
| party_name    | varchar(15)             | YES  |     | NULL    |       |
| serial_number | int(2) unsigned zerofill | NO   | PRI | NULL    |       |
+---------------+-------------------------+------+-----+---------+-------+
8 rows in set (0.022 sec)

MariaDB [politics]> CREATE table legislative_general_election_result (
    -> legislative_general_election_year int(4),
    -> candidate_name char(30),
    -> party_name varchar(15),
    -> candidate_serial_number int(2) zerofill NOT NULL,
    -> seat_amount int,
    -> voting_rank int,
    -> votes_amount int,
    -> FOREIGN KEY (candidate_serial_number)
    -> REFERENCES legislative_candidate(serial_number));
Query OK, 0 rows affected (0.012 sec)

MariaDB [politics]> DESC legislative_general_election_result;
+----------------------------------+-------------------------+------+-----+---------+-------+
| Field                            | Type                    | Null | Key | Default | Extra |
+----------------------------------+-------------------------+------+-----+---------+-------+
| legislative_general_election_year | int(4)                  | YES  |     | NULL    |       |
| candidate_name                   | char(30)                | YES  |     | NULL    |       |
| party_name                       | varchar(15)             | YES  |     | NULL    |       |
| candidate_serial_number          | int(2) unsigned zerofill | NO   | MUL | NULL    |       |
| seat_amount                      | int(11)                 | YES  |     | NULL    |       |
| voting_rank                      | int(11)                 | YES  |     | NULL    |       |
| votes_amount                     | int(11)                 | YES  |     | NULL    |       |
+----------------------------------+-------------------------+------+-----+---------+-------+
7 rows in set (0.050 sec)
```

b) Demonstrate you can insert, update, and delete the data and make explanations
You must insert some data for demonstrate other questions

```
MariaDB [politics]> INSERT INTO province values (32, "West Java");
Query OK, 1 row affected (0.033 sec)

MariaDB [politics]> INSERT INTO province values (12, "North Sumatra");
Query OK, 1 row affected (0.001 sec)

MariaDB [politics]> INSERT INTO province values (64, "East Kalimantan");
Query OK, 1 row affected (0.001 sec)

MariaDB [politics]> INSERT INTO province values (73, "South Sulawesi");
Query OK, 1 row affected (0.001 sec)

MariaDB [politics]> INSERT INTO province values (51, "Bali");
Query OK, 1 row affected (0.001 sec)

MariaDB [politics]> INSERT INTO province values (91, "Papua");
Query OK, 1 row affected (0.001 sec)

MariaDB [politics]> INSERT INTO province values (75, "Gorontalo");
Query OK, 1 row affected (0.001 sec)

MariaDB [politics]> INSERT INTO province values (72, "Central Sulawesi");
Query OK, 1 row affected (0.001 sec)

MariaDB [politics]> INSERT INTO province values (11, "Aceh");
Query OK, 1 row affected (0.001 sec)

MariaDB [politics]> INSERT INTO province values (77, "Bengkulu");
Query OK, 1 row affected (0.001 sec)

MariaDB [politics]> SELECT * FROM province;
+------+-----------------+
| code | province_name   |
+------+-----------------+
|   11 | Aceh            |
|   12 | North Sumatra   |
|   32 | West Java       |
|   51 | Bali            |
|   64 | East Kalimantan |
|   72 | Central Sulawesi|
|   73 | South Sulawesi  |
|   75 | Gorontalo       |
|   77 | Bengkulu        |
|   91 | Papua           |
+------+-----------------+
10 rows in set (0.038 sec)

MariaDB [politics]> INSERT INTO district values (32, 16, "Bekasi");
Query OK, 1 row affected (0.003 sec)

MariaDB [politics]> INSERT INTO district values (12, 04, "Nias");
Query OK, 1 row affected (0.002 sec)

MariaDB [politics]> INSERT INTO district values (64, 01, "Paser");
Query OK, 1 row affected (0.001 sec)

MariaDB [politics]> INSERT INTO district values (73, 17, "Luwu");
Query OK, 1 row affected (0.001 sec)

MariaDB [politics]> INSERT INTO district values (51, 08, "Buleleng");
Query OK, 1 row affected (0.001 sec)

MariaDB [politics]> INSERT INTO district values (91, 22, "Yalimo");
Query OK, 1 row affected (0.001 sec)

MariaDB [politics]> INSERT INTO district values (75, 02, "Boalemo");
Query OK, 1 row affected (0.001 sec)

MariaDB [politics]> SELECT * FROM district;
+---------------+------+---------------+
| province_code | code | district_name |
+---------------+------+---------------+
|            64 |   01 | Paser         |
|            75 |   02 | Boalemo       |
|            12 |   04 | Nias          |
|            51 |   08 | Buleleng      |
|            32 |   16 | Bekasi        |
|            73 |   17 | Luwu          |
|            91 |   22 | Yalimo        |
+---------------+------+---------------+
7 rows in set (0.000 sec)
```

```
MariaDB [politics]> INSERT INTO sub_district values (16, 07, "Cibitung");
Query OK, 1 row affected (0.004 sec)

MariaDB [politics]> INSERT INTO sub_district values (04, 06, "Gido");
Query OK, 1 row affected (0.002 sec)

MariaDB [politics]> INSERT INTO sub_district values (01, 05, "Kuaro");
Query OK, 1 row affected (0.002 sec)

MariaDB [politics]> INSERT INTO sub_district values (17, 02, "Larompong");
Query OK, 1 row affected (0.001 sec)

MariaDB [politics]> INSERT INTO sub_district values (08, 04, "Banjar");
Query OK, 1 row affected (0.001 sec)

MariaDB [politics]> INSERT INTO sub_district values (22, 03, "Abenaho");
Query OK, 1 row affected (0.001 sec)

MariaDB [politics]> INSERT INTO sub_district values (02, 01, "Paguyaman");
Query OK, 1 row affected (0.001 sec)

MariaDB [politics]> SELECT * FROM sub_district;
+---------------+------+-------------------+
| district_code | code | sub_district_name |
+---------------+------+-------------------+
|            02 |   01 | Paguyaman         |
|            17 |   02 | Larompong         |
|            22 |   03 | Abenaho           |
|            08 |   04 | Banjar            |
|            01 |   05 | Kuaro             |
|            04 |   06 | Gido              |
|            16 |   07 | Cibitung          |
+---------------+------+-------------------+
7 rows in set (0.000 sec)

MariaDB [politics]> INSERT INTO village values (07, 712, "Kertamukti");
Query OK, 1 row affected (0.050 sec)

MariaDB [politics]> INSERT INTO village values (06, 620, "Somi");
Query OK, 1 row affected (0.002 sec)

MariaDB [politics]> INSERT INTO village values (05, 513, "Rangan");
Query OK, 1 row affected (0.001 sec)

MariaDB [politics]> INSERT INTO village values (02, 219, "Riwang");
Query OK, 1 row affected (0.001 sec)

MariaDB [politics]> INSERT INTO village values (04, 441, "Tirtasari");
Query OK, 1 row affected (0.001 sec)

MariaDB [politics]> INSERT INTO village values (03, 326, "Halisek");
Query OK, 1 row affected (0.001 sec)

MariaDB [politics]> INSERT INTO village values (01, 108, "Bualo");
Query OK, 1 row affected (0.001 sec)

MariaDB [politics]> SELECT * FROM village;
+-------------------+------+--------------+
| sub_district_code | code | village_name |
+-------------------+------+--------------+
|                01 |  108 | Bualo        |
|                02 |  219 | Riwang       |
|                03 |  326 | Halisek      |
|                04 |  441 | Tirtasari    |
|                05 |  513 | Rangan       |
|                06 |  620 | Somi         |
|                07 |  712 | Kertamukti   |
+-------------------+------+--------------+
7 rows in set (0.000 sec)

MariaDB [politics]> INSERT INTO party values (01, "PKS", 1998, "Ahmad Syaikhu", 5);
Query OK, 1 row affected (0.052 sec)

MariaDB [politics]> INSERT INTO party values (02, "PPP", 1973, "Suharso Monoarfa", 11);
Query OK, 1 row affected (0.004 sec)

MariaDB [politics]> INSERT INTO party values (03, "DEMOKRAT", 2001, "Susilo Bambang Yudhoyono", 4);
Query OK, 1 row affected (0.001 sec)

MariaDB [politics]> INSERT INTO party values (04, "GOLKAR", 1964, "Airlangga Hartarto", 12);
Query OK, 1 row affected (0.001 sec)

MariaDB [politics]> INSERT INTO party values (05, "PDIP", 1973, "Megawati Soekarnoputri", 5);
Query OK, 1 row affected (0.001 sec)

MariaDB [politics]> INSERT INTO party values (06, "NASDEM", 2011, "Surya Paloh", 2);
Query OK, 1 row affected (0.001 sec)

MariaDB [politics]> INSERT INTO party values (07, "GERINDRA", 2008, "Prabowo Subianto", 3);
Query OK, 1 row affected (0.001 sec)

MariaDB [politics]> INSERT INTO party values (08, "GERINDRA", 2008, "Prabowo Subianto", 3);
Query OK, 1 row affected (0.001 sec)

MariaDB [politics]> INSERT INTO party values (09, "PERINDO", 2014, "Hary Tanoesoedibjo", 1);
Query OK, 1 row affected (0.001 sec)

MariaDB [politics]> INSERT INTO party values (10, "PKB", 1998, "Muhaimin Iskandar", 5);
Query OK, 1 row affected (0.001 sec)

MariaDB [politics]> INSERT INTO party values (11, "PAN", 1998, "Zulkifli Hasan", 5);
Query OK, 1 row affected (0.001 sec)
```

```
MariaDB [politics]> SELECT * FROM party;
+------+------------+------------------+-------------------------+-----------------------------------------+
| code | party_name | established_year | party_chairman          | participation_in_legislative_general_election |
+------+------------+------------------+-------------------------+-----------------------------------------+
|   01 | PKS        |             1998 | Ahmad Syaikhu           |                                       5 |
|   02 | PPP        |             1973 | Suharso Monoarfa        |                                      11 |
|   03 | DEMOKRAT   |             2001 | Susilo Bambang Yudhoyono |                                       4 |
|   04 | GOLKAR     |             1964 | Airlangga Hartarto      |                                      12 |
|   05 | PDIP       |             1973 | Megawati Soekarnoputri  |                                       5 |
|   06 | NASDEM     |             2011 | Surya Paloh             |                                       2 |
|   07 | GERINDRA   |             2008 | Prabowo Subianto        |                                       3 |
|   08 | GERINDRA   |             2008 | Prabowo Subianto        |                                       3 |
|   09 | PERINDO    |             2014 | Hary Tanoesoedibjo      |                                       1 |
|   10 | PKB        |             1998 | Muhaimin Iskandar       |                                       5 |
|   11 | PAN        |             1998 | Zulkifli Hasan          |                                       5 |
+------+------------+------------------+-------------------------+-----------------------------------------+
11 rows in set (0.000 sec)

MariaDB [politics]> INSERT INTO legislative_candidate values (3216070911780003, "Bambang Purwanto", "Male", "Kertamukti Street Block A-01, Cibitung", 712, 01, "PKS", 24);
Query OK, 1 row affected (0.033 sec)

MariaDB [politics]> INSERT INTO legislative_candidate values (1204064812670002, "Dewi Zebua", "Female", "2B Somi Street, Gido", 620, 05, "PDIP", 04);
Query OK, 1 row affected (0.002 sec)

MariaDB [politics]> INSERT INTO legislative_candidate values (6401055210570001, "Arlina", "Female", "Rangan Village block I-47, Kuaro", 513, 03, "DEMOKRAT", 31);
Query OK, 1 row affected (0.002 sec)

MariaDB [politics]> INSERT INTO legislative_candidate values (7317026403800004, "Herawati", "Female", "Riwang Housing number 6, Larompong", 219, 04, "GOLKAR", 02);
Query OK, 1 row affected (0.001 sec)

MariaDB [politics]> INSERT INTO legislative_candidate values (5108041708820005, "Nyoman Tirtawan", "Male", "18E Tirtasari Village, Banjar", 441, 06, "NASDEM", 05);
Query OK, 1 row affected (0.001 sec)

MariaDB [politics]> INSERT INTO legislative_candidate values (9122030305750003, "Elia Yare", "Male", "Halisek Street number 5, Abenaho", 326, 07, "GERINDRA", 06);
Query OK, 1 row affected (0.001 sec)

MariaDB [politics]> INSERT INTO legislative_candidate values (7502017007900006, "Sarippa Atute", "Female", "Bualo Housing Block D-09, Paguyaman", 108, 02, "PPP", 01);
Query OK, 1 row affected (0.001 sec)

MariaDB [politics]> SELECT * FROM legislative_candidate;
+------------------+-----------------+--------+-----------------------------------+--------------+------------+------------+---------------+
| ID_card_number   | candidate_name  | gender | address                           | village_code | party_code | party_name | serial_number |
+------------------+-----------------+--------+-----------------------------------+--------------+------------+------------+---------------+
| 7502017007900006 | Sarippa Atute   | Female | Bualo Housing Block D-09, Paguyaman |          108 |         02 | PPP        |            01 |
| 7317026403800004 | Herawati        | Female | Riwang Housing number 6, Larompong |          219 |         04 | GOLKAR     |            02 |
| 1204064812670002 | Dewi Zebua      | Female | 2B Somi Street, Gido              |          620 |         05 | PDIP       |            04 |
| 5108041708820005 | Nyoman Tirtawan | Male   | 18E Tirtasari Village, Banjar     |          441 |         06 | NASDEM     |            05 |
| 9122030305750003 | Elia Yare       | Male   | Halisek Street number 5, Abenaho  |          326 |         07 | GERINDRA   |            06 |
| 3216070911780003 | Bambang Purwanto | Male  | Kertamukti Street Block A-01, Cibitung |      712 |         01 | PKS        |            24 |
| 6401055210570001 | Arlina          | Female | Rangan Village block I-47, Kuaro  |          513 |         03 | DEMOKRAT   |            31 |
+------------------+-----------------+--------+-----------------------------------+--------------+------------+------------+---------------+
7 rows in set (0.000 sec)

MariaDB [politics]> INSERT INTO legislative_general_election_result values (2019, "Nyoman Tirtawan", "NASDEM", 05, 59, 5, 12661792);
Query OK, 1 row affected (0.053 sec)

MariaDB [politics]> INSERT INTO legislative_general_election_result values (2009, "Arlina", "DEMOKRAT", 31, 150, 1, 21703137);
Query OK, 1 row affected (0.002 sec)

MariaDB [politics]> INSERT INTO legislative_general_election_result values (1997, "Herawati", "GOLKAR", 02, 325, 1, 84187907);
Query OK, 1 row affected (0.002 sec)

MariaDB [politics]> INSERT INTO legislative_general_election_result values (1992, "Sarippa Atute", "PPP", 01, 62, 2, 16624647);
Query OK, 1 row affected (0.001 sec)

MariaDB [politics]> INSERT INTO legislative_general_election_result values (2014, "Elia Yare", "GERINDRA", 06, 73, 3, 14760371);
Query OK, 1 row affected (0.001 sec)

MariaDB [politics]> INSERT INTO legislative_general_election_result values (1999, "Bambang Purwanto", "PKS", 24, 7, 7, 1436565);
Query OK, 1 row affected (0.001 sec)

MariaDB [politics]> INSERT INTO legislative_general_election_result values (2014, "Dewi Zebua", "PDIP", 04, 109, 2, 21026629);
Query OK, 1 row affected (0.001 sec)

MariaDB [politics]> SELECT * FROM legislative_general_election_result;
+----------------------------------+-----------------+------------+-------------------------+-------------+-------------+--------------+
| legislative_general_election_year | candidate_name  | party_name | candidate_serial_number | seat_amount | voting_rank | votes_amount |
+----------------------------------+-----------------+------------+-------------------------+-------------+-------------+--------------+
|                             2019 | Nyoman Tirtawan | NASDEM     |                      05 |          59 |           5 |     12661792 |
|                             2009 | Arlina          | DEMOKRAT   |                      31 |         150 |           1 |     21703137 |
|                             1997 | Herawati        | GOLKAR     |                      02 |         325 |           1 |     84187907 |
|                             1992 | Sarippa Atute   | PPP        |                      01 |          62 |           2 |     16624647 |
|                             2014 | Elia Yare       | GERINDRA   |                      06 |          73 |           3 |     14760371 |
|                             1999 | Bambang Purwanto | PKS       |                      24 |           7 |           7 |      1436565 |
|                             2014 | Dewi Zebua      | PDIP       |                      04 |         109 |           2 |     21026629 |
+----------------------------------+-----------------+------------+-------------------------+-------------+-------------+--------------+
7 rows in set (0.000 sec)
```

The **INSERT INTO** statement is used to insert new records in a table. If we are adding values for all the columns of the table, we do not need to specify the column names in the query. However, make sure the order of the values is in the same order as the columns in the table.

```
MariaDB [politics]> SELECT * FROM party;
+------+------------+-----------------+-------------------------+-----------------------------------------+
| code | party_name | established_year | party_chairman          | participation_in_legislative_general_election |
+------+------------+-----------------+-------------------------+-----------------------------------------+
|   01 | PKS        |            1998 | Ahmad Syaikhu           |                                       5 |
|   02 | PPP        |            1973 | Suharso Monoarfa        |                                      11 |
|   03 | DEMOKRAT   |            2001 | Susilo Bambang Yudhoyono |                                      4 |
|   04 | GOLKAR     |            1964 | Airlangga Hartarto      |                                      12 |
|   05 | PDIP       |            1973 | Megawati Soekarnoputri  |                                       5 |
|   06 | NASDEM     |            2011 | Surya Paloh             |                                       2 |
|   07 | GERINDRA   |            2008 | Prabowo Subianto        |                                       3 |
|   08 | GERINDRA   |            2008 | Prabowo Subianto        |                                       3 |
|   09 | PERINDO    |            2014 | Hary Tanoesoedibjo      |                                       1 |
|   10 | PKB        |            1998 | Muhaimin Iskandar       |                                       5 |
|   11 | PAN        |            1998 | Zulkifli Hasan          |                                       5 |
+------+------------+-----------------+-------------------------+-----------------------------------------+
11 rows in set (0.001 sec)

MariaDB [politics]> UPDATE party
    -> SET party_chairman = "Agus Harimurti Yudhoyono"
    -> WHERE party_name = "DEMOKRAT";
Query OK, 1 row affected (0.045 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [politics]> SELECT * FROM party;
+------+------------+-----------------+-------------------------+-----------------------------------------+
| code | party_name | established_year | party_chairman          | participation_in_legislative_general_election |
+------+------------+-----------------+-------------------------+-----------------------------------------+
|   01 | PKS        |            1998 | Ahmad Syaikhu           |                                       5 |
|   02 | PPP        |            1973 | Suharso Monoarfa        |                                      11 |
|   03 | DEMOKRAT   |            2001 | Agus Harimurti Yudhoyono |                                      4 |
|   04 | GOLKAR     |            1964 | Airlangga Hartarto      |                                      12 |
|   05 | PDIP       |            1973 | Megawati Soekarnoputri  |                                       5 |
|   06 | NASDEM     |            2011 | Surya Paloh             |                                       2 |
|   07 | GERINDRA   |            2008 | Prabowo Subianto        |                                       3 |
|   08 | GERINDRA   |            2008 | Prabowo Subianto        |                                       3 |
|   09 | PERINDO    |            2014 | Hary Tanoesoedibjo      |                                       1 |
|   10 | PKB        |            1998 | Muhaimin Iskandar       |                                       5 |
|   11 | PAN        |            1998 | Zulkifli Hasan          |                                       5 |
+------+------------+-----------------+-------------------------+-----------------------------------------+
11 rows in set (0.000 sec)
```

The **UPDATE** statement is used to modify the existing records in a table and the example that I made; I changed the chairman of the DEMOKRAT party from Susilo Bambang Yudhoyono to Agus Harimurti Yudhoyono.

```
MariaDB [politics]> SELECT * FROM party;
+------+------------+------------------+--------------------------+-----------------------------------------------+
| code | party_name | established_year | party_chairman           | participation_in_legislative_general_election |
+------+------------+------------------+--------------------------+-----------------------------------------------+
|   01 | PKS        |             1998 | Ahmad Syaikhu            |                                             5 |
|   02 | PPP        |             1973 | Suharso Monoarfa         |                                            11 |
|   03 | DEMOKRAT   |             2001 | Agus Harimurti Yudhoyono |                                             4 |
|   04 | GOLKAR     |             1964 | Airlangga Hartarto       |                                            12 |
|   05 | PDIP       |             1973 | Megawati Soekarnoputri   |                                             5 |
|   06 | NASDEM     |             2011 | Surya Paloh              |                                             2 |
|   07 | GERINDRA   |             2008 | Prabowo Subianto         |                                             3 |
|   08 | GERINDRA   |             2008 | Prabowo Subianto         |                                             3 |
|   09 | PERINDO    |             2014 | Hary Tanoesoedibjo       |                                             1 |
|   10 | PKB        |             1998 | Muhaimin Iskandar        |                                             5 |
|   11 | PAN        |             1998 | Zulkifli Hasan           |                                             5 |
+------+------------+------------------+--------------------------+-----------------------------------------------+
11 rows in set (0.000 sec)

MariaDB [politics]> DELETE from party
    -> WHERE code = 08;
Query OK, 1 row affected (0.022 sec)

MariaDB [politics]> SELECT * FROM party;
+------+------------+------------------+--------------------------+-----------------------------------------------+
| code | party_name | established_year | party_chairman           | participation_in_legislative_general_election |
+------+------------+------------------+--------------------------+-----------------------------------------------+
|   01 | PKS        |             1998 | Ahmad Syaikhu            |                                             5 |
|   02 | PPP        |             1973 | Suharso Monoarfa         |                                            11 |
|   03 | DEMOKRAT   |             2001 | Agus Harimurti Yudhoyono |                                             4 |
|   04 | GOLKAR     |             1964 | Airlangga Hartarto       |                                            12 |
|   05 | PDIP       |             1973 | Megawati Soekarnoputri   |                                             5 |
|   06 | NASDEM     |             2011 | Surya Paloh              |                                             2 |
|   07 | GERINDRA   |             2008 | Prabowo Subianto         |                                             3 |
|   09 | PERINDO    |             2014 | Hary Tanoesoedibjo       |                                             1 |
|   10 | PKB        |             1998 | Muhaimin Iskandar        |                                             5 |
|   11 | PAN        |             1998 | Zulkifli Hasan           |                                             5 |
+------+------------+------------------+--------------------------+-----------------------------------------------+
10 rows in set (0.000 sec)
```

The **DELETE** statement is used to delete existing records in a table and the example that I made; I delete the row where the party code is 8 which is a GERINDRA party because it has the same values as the party code 7.

c) Demonstrate you can implement **JOIN, LEFT JOIN, and RIGHT JOIN**

```
MariaDB [politics]> SELECT c.candidate_name, c.party_name,
    -> c.serial_number, c.address, v.village_name,
    -> s.sub_district_name, d.district_name, pr.province_name
    -> FROM legislative_candidate c
    -> JOIN village v ON v.code = c.village_code
    -> JOIN sub_district s ON s.code = v.sub_district_code
    -> JOIN district d ON d.code = s.district_code
    -> JOIN province pr ON pr.code = d.province_code;
+-----------------+------------+---------------+-----------------------------------------+--------------+-----------------+---------------+---------------+
| candidate_name  | party_name | serial_number | address                                 | village_name | sub_district_name | district_name | province_name |
+-----------------+------------+---------------+-----------------------------------------+--------------+-----------------+---------------+---------------+
| Sarippa Atute   | PPP        |            01 | Bualo Housing Block D-09, Paguyaman     | Bualo        | Paguyaman       | Boalemo       | Gorontalo     |
| Herawati        | GOLKAR     |            02 | Riwang Housing number 6, Larompong      | Riwang       | Larompong       | Luwu          | South Sulawesi |
| Dewi Zebua      | PDIP       |            04 | 2B Somi Street, Gido                    | Somi         | Gido            | Nias          | North Sumatra |
| Nyoman Tirtawan | NASDEM     |            05 | 18E Tirtasari Village, Banjar           | Tirtasari    | Banjar          | Buleleng      | Bali          |
| Elia Yare       | GERINDRA   |            06 | Halisek Street number 5, Abenaho        | Halisek      | Abenaho         | Yalimo        | Papua         |
| Bambang Purwanto | PKS       |            24 | Kertamukti Street Block A-01, Cibitung  | Kertamukti   | Cibitung        | Bekasi        | West Java     |
| Arlina          | DEMOKRAT   |            31 | Rangan Village block I-47, Kuaro        | Rangan       | Kuaro           | Paser         | East Kalimantan |
+-----------------+------------+---------------+-----------------------------------------+--------------+-----------------+---------------+---------------+
7 rows in set (0.038 sec)
```

A **JOIN** clause is used to combine rows from two or more tables, based on a related column between them. For **JOIN or (INNER) JOIN** its returns records that have matching values in both tables and the example that I made, I displayed all the legislative candidate address include their address, village, sub-district, district, and province along with their name and their name of their party and their serial number and all their data is in a different table.

```
MariaDB [politics]> SELECT p.party_name, c.candidate_name, c.serial_number
    -> FROM party p
    -> LEFT JOIN legislative_candidate c ON p.code = c.party_code;
+------------+------------------+---------------+
| party_name | candidate_name   | serial_number |
+------------+------------------+---------------+
| PKS        | Bambang Purwanto |            24 |
| PPP        | Sarippa Atute    |            01 |
| DEMOKRAT   | Arlina           |            31 |
| GOLKAR     | Herawati         |            02 |
| PDIP       | Dewi Zebua       |            04 |
| NASDEM     | Nyoman Tirtawan  |            05 |
| GERINDRA   | Elia Yare        |            06 |
| PERINDO    | NULL             |          NULL |
| PKB        | NULL             |          NULL |
| PAN        | NULL             |          NULL |
+------------+------------------+---------------+
10 rows in set (0.000 sec)
```

For **LEFT (OUTER) JOIN** it returns all records from the left table (table1), and the matching records from the right table (table2). The result is 0 records from the right side, if there is no match and the example that I made, I displayed all the party name from party tables along with the candidate name and candidate serial number in table legislative candidate and some party name return NULL values for the candidate name and candidate serial number because it doesn't match with any values in the legislative candidate table which is the right-side table.

```
MariaDB [politics]> SELECT c.candidate_name, c.party_name, c.address,
    -> v.village_name, s.sub_district_name,
    -> d.district_name, pr.province_name
    -> FROM legislative_candidate c
    -> JOIN village v ON v.code = c.village_code
    -> JOIN sub_district s ON s.code = v.sub_district_code
    -> JOIN district d ON d.code = s.district_code
    -> RIGHT JOIN province pr ON pr.code = d.province_code
    -> ORDER BY province_name ASC;
+------------------+------------+-----------------------------------+--------------+-----------------+---------------+------------------+
| candidate_name   | party_name | address                           | village_name | sub_district_name | district_name | province_name    |
+------------------+------------+-----------------------------------+--------------+-----------------+---------------+------------------+
| NULL             | NULL       | NULL                              | NULL         | NULL            | NULL          | Aceh             |
| Nyoman Tirtawan  | NASDEM     | 18E Tirtasari Village, Banjar     | Tirtasari    | Banjar          | Buleleng      | Bali             |
| NULL             | NULL       | NULL                              | NULL         | NULL            | NULL          | Bengkulu         |
| NULL             | NULL       | NULL                              | NULL         | NULL            | NULL          | Central Sulawesi |
| Arlina           | DEMOKRAT   | Rangan Village block I-47, Kuaro  | Rangan       | Kuaro           | Paser         | East Kalimantan  |
| Sarippa Atute    | PPP        | Bualo Housing Block D-09, Paguyaman | Bualo      | Paguyaman       | Boalemo       | Gorontalo        |
| Dewi Zebua       | PDIP       | 2B Somi Street, Gido              | Somi         | Gido            | Nias          | North Sumatra    |
| Elia Yare        | GERINDRA   | Halisek Street number 5, Abenaho  | Halisek      | Abenaho         | Yalimo        | Papua            |
| Herawati         | GOLKAR     | Riwang Housing number 6, Larompong | Riwang      | Larompong       | Luwu          | South Sulawesi   |
| Bambang Purwanto | PKS        | Kertamukti Street Block A-01, Cibitung | Kertamukti | Cibitung      | Bekasi        | West Java        |
+------------------+------------+-----------------------------------+--------------+-----------------+---------------+------------------+
10 rows in set (0.001 sec)
```

For **RIGHT (OUTER) JOIN** it returns all records from the right table (table2), and the matching records from the left table (table1). The result is 0 records from the left side, if there is no match and the example that I made, I displayed all the legislative candidate address include their address, village, sub-district, district, and province along with the name of their name and party, and some province return NULL values for the candidate name, party name, address, village, sub-district, and district because it doesn't match with any values in the legislative candidate, village, sub-district, and district table which is the left-side table.

d) Demonstrate you can use 3 out of 5 aggregate functions

```
MariaDB [politics]> SELECT MAX(votes_amount) AS max_votes_amount,
    -> MIN(votes_amount) AS min_votes_amount,
    -> SUM(votes_amount) AS sum_of_votes_amount,
    -> AVG(votes_amount) AS avg_of_votes_amount
    -> FROM legislative_general_election_result;
+------------------+------------------+--------------------+--------------------+
| max_votes_amount | min_votes_amount | sum_of_votes_amount | avg_of_votes_amount |
+------------------+------------------+--------------------+--------------------+
|         84187907 |          1436565 |          172401048 |       24628721.1429 |
+------------------+------------------+--------------------+--------------------+
1 row in set (0.003 sec)
```

The **MIN()** function returns the smallest value of the selected column, the **MAX()** function returns the largest value of the selected column, the **AVG()** function returns the average value of a numeric column and the **SUM()** function returns the total sum of a numeric column, and the example that I made, I displayed the max and min values for the amount of the votes and also the sum and the average of the amount of the votes in the legislative general election result table.

```
MariaDB [politics]> SELECT established_year AS party_established_year,
    -> COUNT(established_year) AS party_amount
    -> FROM party
    -> GROUP BY established_year;
+------------------------+--------------+
| party_established_year | party_amount |
+------------------------+--------------+
|                   1964 |            1 |
|                   1973 |            2 |
|                   1998 |            3 |
|                   2001 |            1 |
|                   2008 |            1 |
|                   2011 |            1 |
|                   2014 |            1 |
+------------------------+--------------+
7 rows in set (0.030 sec)
```

The **COUNT()** function returns the number of rows that matches a specified criterion and the example that I made, I displayed the number of parties that were established in a given year and group the number of parties by the year they were established.

```
MariaDB [politics]> SELECT voting_rank AS party_voting_rank,
    -> COUNT(voting_rank) AS party_amount
    -> FROM legislative_general_election_result
    -> WHERE voting_rank < 6
    -> GROUP BY voting_rank;
+-------------------+--------------+
| party_voting_rank | party_amount |
+-------------------+--------------+
|                 1 |            2 |
|                 2 |            2 |
|                 3 |            1 |
|                 5 |            1 |
+-------------------+--------------+
4 rows in set (0.000 sec)
```

For the last example is I used **condition** and I displayed the number of parties that are ranked in voting with the highest-ranking condition or above 6th rank (a condition with a ranking score of less than 6 as the highest rank) and group the number of parties by the voting ranking obtained.

e) Demonstrate you can use subquery (nested query) and sub-subquery

```
MariaDB [politics]> SELECT * FROM legislative_general_election_result
    -> WHERE seat_amount >
    -> (SELECT AVG(seat_amount)
    -> FROM legislative_general_election_result);
+---------------------------------+----------------+------------+------------------------+-------------+-------------+--------------+
| legislative_general_election_year | candidate_name | party_name | candidate_serial_number | seat_amount | voting_rank | votes_amount |
+---------------------------------+----------------+------------+------------------------+-------------+-------------+--------------+
|                            2009 | Arlina         | DEMOKRAT   |                     31 |         150 |           1 |     21703137 |
|                            1997 | Herawati       | GOLKAR     |                     02 |         325 |           1 |     84187907 |
+---------------------------------+----------------+------------+------------------------+-------------+-------------+--------------+
2 rows in set (0.020 sec)
```

A **Subquery** or Inner query or a Nested query is a query within another SQL query and embedded within the WHERE clause. A subquery is used to return data that will be used in the main query as a condition to further restrict the data to be retrieved, and the example that I made, I displayed all data in the legislative general election result table which has the amount of the seats more than the average amount of the seats in the legislative general election result table.

```
MariaDB [politics]> SELECT * FROM party
    -> WHERE established_year < ANY (
    -> SELECT MIN(legislative_general_election_year)
    -> FROM legislative_general_election_result
    -> WHERE party_name IN (
    -> SELECT party_name
    -> FROM legislative_candidate
    -> WHERE serial_number BETWEEN 1 AND 5));
+------+------------+-----------------+----------------------+-----------------------------------------+
| code | party_name | established_year | party_chairman       | participation_in_legislative_general_election |
+------+------------+-----------------+----------------------+-----------------------------------------+
|   02 | PPP        |            1973 | Suharso Monoarfa     |                                      11 |
|   04 | GOLKAR     |            1964 | Airlangga Hartarto   |                                      12 |
|   05 | PDIP       |            1973 | Megawati Soekarnoputri |                                     5 |
+------+------------+-----------------+----------------------+-----------------------------------------+
3 rows in set (0.000 sec)
```

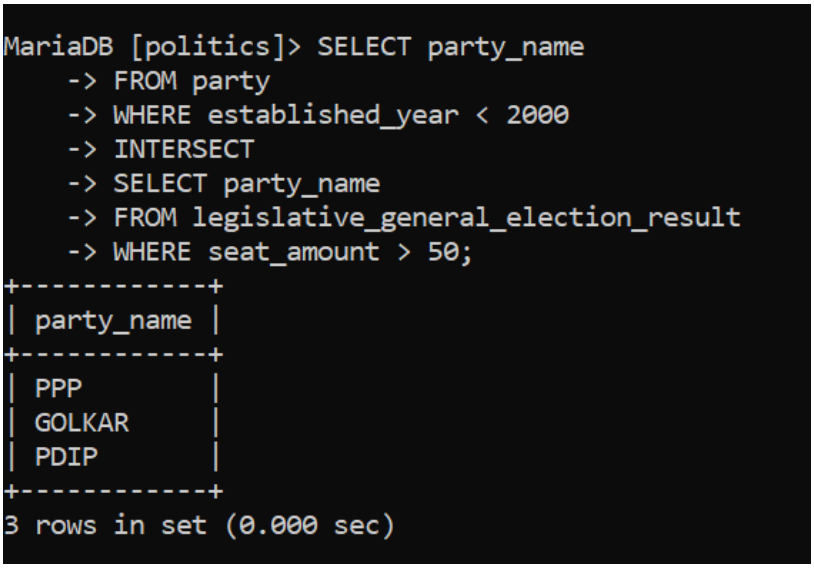
A **nested subquery** is a subquery nested within another subquery. There is no limit to the level of subquery nesting you can define, however, queries with three or more levels take considerably longer to run than do smaller queries. A subquery can be nested inside other subqueries. SQL has an ability to nest queries within one another. A subquery is a SELECT statement that is nested within another SELECT statement and which return intermediate results. SQL executes innermost subquery first, then next level, and the example that I made, I displayed all data in the party table where the established year was smaller than the minimum year of the legislative general election year in the legislative general election results table where the party has a legislative candidate serial number between 1 and 5.

f) Demonstrate you can implement UNION and INTERSECTION

```
MariaDB [politics]> SELECT party_name
    -> FROM party
    -> WHERE participation_in_legislative_general_election > 4
    -> UNION
    -> SELECT party_name
    -> FROM legislative_general_election_result
    -> WHERE voting_rank < 6;
+------------+
| party_name |
+------------+
| PKS        |
| PPP        |
| GOLKAR     |
| PDIP       |
| PKB        |
| PAN        |
| NASDEM     |
| DEMOKRAT   |
| GERINDRA   |
+------------+
9 rows in set (0.002 sec)
```

The **UNION** operator is used to combine the result-set of two or more SELECT statements. Every SELECT statement within UNION must have the same number of columns, the

columns must also have similar data types, and the columns in every SELECT statement must also be in the same order, and the example that I made, I combine the names of parties in the party table that participated in the legislative general elections more than 4 times with the names of the parties in the legislative general election results table which have a voting rating above 6th rank (voting rank less than 6).



An **INTERSECT** query returns the intersection of 2 or more datasets. If a record exists in both data sets, it will be included in the INTERSECT results. However, if a record exists in one data set and not in the other, it will be omitted from the INTERSECT results and there must be same number of expressions in both SELECT statements and have similar data types, and the example that I made, I combine the names of the parties in the party table which was established under 2000 (the year it was established is smaller than 2000) which intersects with the names of the parties in the legislative general election result table which has more than 50 amounts of the seat.

g) Demonstrate you can implement modify/change the column

```
MariaDB [politics]> DESC party;
+-------------------------------------------+-------------------------+------+-----+---------+-------+
| Field                                     | Type                    | Null | Key | Default | Extra |
+-------------------------------------------+-------------------------+------+-----+---------+-------+
| code                                      | int(2) unsigned zerofill | NO   | PRI | NULL    |       |
| party_name                                | varchar(15)             | YES  |     | NULL    |       |
| established_year                          | int(4)                  | YES  |     | NULL    |       |
| party_chairman                            | char(30)                | YES  |     | NULL    |       |
| participation_in_legislative_general_election | int(11)                 | YES  |     | NULL    |       |
+-------------------------------------------+-------------------------+------+-----+---------+-------+
5 rows in set (0.038 sec)

MariaDB [politics]> ALTER TABLE party
    -> MODIFY party_chairman varchar(50)
    -> AFTER party_name;
Query OK, 10 rows affected (0.038 sec)
Records: 10  Duplicates: 0  Warnings: 0

MariaDB [politics]> DESC party;
+-------------------------------------------+-------------------------+------+-----+---------+-------+
| Field                                     | Type                    | Null | Key | Default | Extra |
+-------------------------------------------+-------------------------+------+-----+---------+-------+
| code                                      | int(2) unsigned zerofill | NO   | PRI | NULL    |       |
| party_name                                | varchar(15)             | YES  |     | NULL    |       |
| party_chairman                            | varchar(50)             | YES  |     | NULL    |       |
| established_year                          | int(4)                  | YES  |     | NULL    |       |
| participation_in_legislative_general_election | int(11)                 | YES  |     | NULL    |       |
+-------------------------------------------+-------------------------+------+-----+---------+-------+
5 rows in set (0.003 sec)
```

The **MODIFY** clause is to change the data type and also the position of a column in a table and the example that I made, I modified the data type of the party chairman column in the party table from char(30) to varchar(50) and also, I modified the position of the column to the after the party name column.

```
MariaDB [politics]> DESC legislative_general_election_result;
+-----------------------------------+-------------------------+------+-----+---------+-------+
| Field                             | Type                    | Null | Key | Default | Extra |
+-----------------------------------+-------------------------+------+-----+---------+-------+
| legislative_general_election_year | int(4)                  | YES  |     | NULL    |       |
| candidate_name                    | char(30)                | YES  |     | NULL    |       |
| party_name                        | varchar(15)             | YES  |     | NULL    |       |
| candidate_serial_number           | int(2) unsigned zerofill | NO   | MUL | NULL    |       |
| seat_amount                       | int(11)                 | YES  |     | NULL    |       |
| voting_rank                       | int(11)                 | YES  |     | NULL    |       |
| votes_amount                      | int(11)                 | YES  |     | NULL    |       |
+-----------------------------------+-------------------------+------+-----+---------+-------+
7 rows in set (0.022 sec)

MariaDB [politics]> ALTER TABLE legislative_general_election_result
    -> CHANGE candidate_name one_of_the_candidate_names varchar(50);
Query OK, 7 rows affected (0.045 sec)
Records: 7  Duplicates: 0  Warnings: 0

MariaDB [politics]> DESC legislative_general_election_result;
+-----------------------------------+-------------------------+------+-----+---------+-------+
| Field                             | Type                    | Null | Key | Default | Extra |
+-----------------------------------+-------------------------+------+-----+---------+-------+
| legislative_general_election_year | int(4)                  | YES  |     | NULL    |       |
| one_of_the_candidate_names        | varchar(50)             | YES  |     | NULL    |       |
| party_name                        | varchar(15)             | YES  |     | NULL    |       |
| candidate_serial_number           | int(2) unsigned zerofill | NO   | MUL | NULL    |       |
| seat_amount                       | int(11)                 | YES  |     | NULL    |       |
| voting_rank                       | int(11)                 | YES  |     | NULL    |       |
| votes_amount                      | int(11)                 | YES  |     | NULL    |       |
+-----------------------------------+-------------------------+------+-----+---------+-------+
7 rows in set (0.006 sec)
```

The **CHANGE** clause offers important additions to the renaming process. It can be used to rename a column and change the data type of that column with the same command and the example that I made, I changed the name of the candidate name column in the legislative general election result table to one of the candidate names and also changed the column data type from char (30) to varchar (50) in one command line.